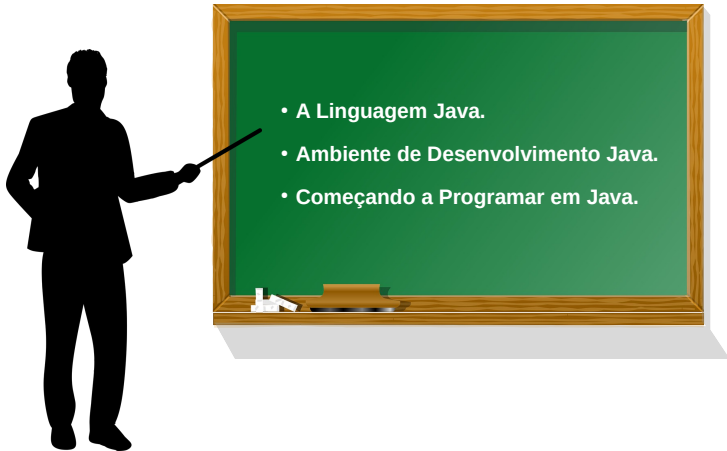


Introdução à Linguagem Java

Luiz Henrique de Campos Merschmann
Departamento de Computação Aplicada
Universidade Federal de Lavras

luiz.hcm@ufla.br

Na Aula de Hoje



A Linguagem Java

- ▶ O objetivo do *Java* é possibilitar a escrita de programas que possam ser executados em uma grande variedade de sistemas computacionais e dispositivos controlados por computador.
- ▶ Segredo?
 - ▶ Conceito de Máquina Virtual (JVM)!
- ▶ Atualmente Java é utilizada para:
 - ▶ Desenvolver aplicativos corporativos de grande porte.
 - ▶ Aprimorar a funcionalidade de servidores da web.
 - ▶ Fornecer aplicativos para dispositivos voltados ao consumo popular.
 - ▶ Desenvolvimento de aplicativos Android (*smartphones* e *tablets*).

Ambiente de Desenvolvimento Java

Passos para criação e execução de um aplicativo Java:

- ▶ Editar.
- ▶ Compilar.
- ▶ Executar.

Ambiente de Desenvolvimento Java

Editando um programa

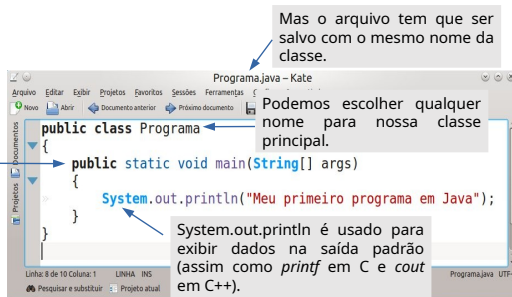
- ▶ Editar um arquivo usando um **programa editor**.
 - ▶ Utilizando editores conhecidos, tais como *vi* e *emacs* (Linux) e *Bloco de Notas* (Windows).
 - ▶ Utilizando Ambientes de Desenvolvimento Integrado (*IDEs*), tais como, *Eclipse*, *NetBeans*, *VSCode* etc.



Ambiente de Desenvolvimento Java

Editando um programa

Assim como em C/C++ um programa começa com *int main*, em Java a classe principal do programa precisa ter um método main com essa assinatura.



Ambiente de Desenvolvimento Java

Compilando um programa

- ▶ Ao compilar um programa em Java, ao invés de se gerar um executável para um sistema específico (como é o caso de *C/C++*), é gerado um arquivo chamado **bytecode** (**.class**) que pode ser executado por uma **Máquina Virtual Java** (**JVM**).



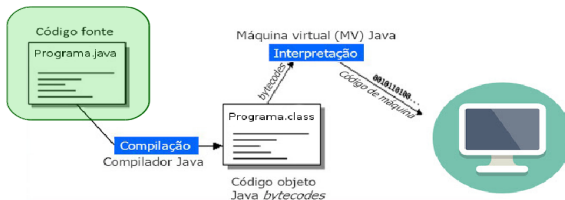
- ▶ Qual a vantagem disso?



A grande vantagem é que você pode compilar seu código uma única vez e o **bytecode** gerado pode ser executado em qualquer sistema que tenha a JVM.

Ambiente de Desenvolvimento Java

Compilando um programa



- ▶ Tendo em mãos o código fonte, o próximo passo é compilá-lo para obtermos o **bytecode** que poderá ser usado pela JVM.
- ▶ Compilando a partir de um terminal:

```
luiz@HP:~$ javac Programa.java
```

`javac` é o compilador Java

Você passa como parâmetro o arquivo com o código-fonte que quer compilar.

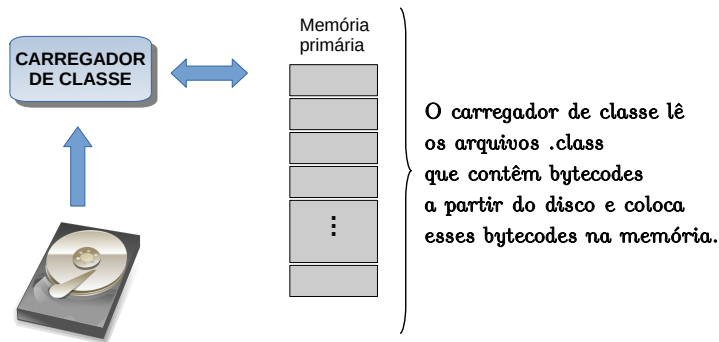
É gerado então o arquivo .class que pode ser executado pela máquina virtual Java (JVM).

```
Mai 14 16:27 Programa.class  
Mai 14 10:49 Programa.java
```


Ambiente de Desenvolvimento Java

Executando um programa

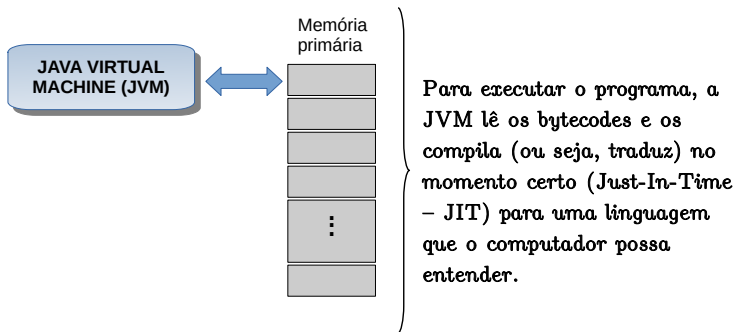
- ▶ Antes de executar um programa a *JVM* armazena o programa na memória (**carregamento**).



Ambiente de Desenvolvimento Java

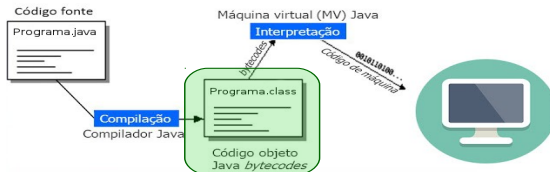
Executando um programa

- ▶ Em seguida, a *JVM* executa os **bytecodes** do programa, ou seja, realiza as instruções especificadas por ele.
 - ▶ Nas primeiras versões, a *JVM* era apenas um interpretador para bytecodes (execução lenta).
 - ▶ Atualmente as *JVMs* utilizam uma combinação de interpretação e compilação *just in time* (*JIT*) (execução mais rápida).



Ambiente de Desenvolvimento Java

Executando um programa



- ▶ Tendo em mãos o **bytecode**, o próximo passo é fazer a *JVM* executar nosso programa.
- ▶ Executando a partir de um terminal:

Usamos o comando `java` para executar o programa

```
luiz@HP:~$ java Programa  
Meu primeiro programa em Java
```

E aí está a execução do nosso primeiro programal

Você passa como parâmetro o nome da classe principal, ou seja, o nome do arquivo `.class`, sem a extensão.

Começando a Programar em Java

Tipos primitivos e estruturas básicas

- ▶ Os tipos primitivos em Java são similares aos de C/C++:
 - ▶ *int*, *float*, *double*.
- ▶ O tipo booleano é denominado *boolean*.
- ▶ As strings não são tipos primitivos, são objetos. A classe é a *String*.
- ▶ Os operadores aritméticos são iguais aos do C/C++:
 - ▶ `+` `-` `*` `/` `%` `+=` `-=` `++` `--`
- ▶ Os operadores lógicos também são iguais aos do C/C++:
 - ▶ `==` `!=` `&&` `||` `!`
- ▶ As sintaxes das estruturas condicionais (*if*, *if...else*) e de repetição (*for*, *while*) são iguais às do C/C++.
- ▶ Por fim, assim como em C/C++, comentários são inseridos utilizando-se: `//` ou `/* ... */`

Nosso Primeiro Programa

```
// Programa de adição que exibe a soma de dois números
import java.util.Scanner; //programa utiliza a classe Scanner
public class Adicao{
    // Método principal inicia a execução do aplicativo Java
    public static void main(String[] args){
        // Cria um Scanner para obter entrada de dados na janela de comando
        Scanner entrada = new Scanner(System.in);

        int numero1; // Primeiro número
        int numero2; // Segundo número
        int soma; //Soma de numero1 e numero2

        System.out.print("Entre com o primeiro numero: "); // Prompt
        numero1 = entrada.nextInt(); // Lê do primeiro número fornecido pelo usuário

        System.out.print("Entre com o segundo numero: "); // Prompt
        numero2 = entrada.nextInt(); // Lê do segundo número fornecido pelo usuário

        soma = numero1 + numero2; // Soma os números e armazena o resultado em soma

        System.out.printf("A soma eh igual a %d\n", soma); // Exibe a soma
    } //fim do método main
} //fim da classe Adicao
```

Importa a classe Scanner para uso neste programa.

Variáveis que são declaradas mas não inicializadas.

Perguntas?

